AD-A067 602 WHARTON SCHOOL PHILADELPHIA PA DEPT OF DECISION SCIENCES F/G 12/1
A SPACE EFFICIENT DYNAMIC ALLOCATION ALGORITHM FOR QUEUEING MES--ETC(U)
JAN 79 E BEYER, P BUNEMAN N00014-75-C-0440
UNCLASSIFIED 79-01-04 NL

END
DATE FILMED
6-79
DDC

Wharton

Department of Decision Sciences

University of
Pennsylvania
Philadelphia PA 19104

A Space Efficient Dynamic Allocation Algorithm for Queueing Messages

Eric Beyer
Peter Buneman

79-01-04

(12)

Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104

January 23, 1979

Supported in part by ONR Contract Number N00014-75-C-0440.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 79-01-04
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): A SPACE EFFICIENT DYNAMIC ALLOCATION ALGORITHM FOR QUEUEING MESSAGES.

5. TYPE OF REPORT & PERIOD COVERED: Technical Report,

6. PERFORMING ORG. REPORT NUMBER: 79-01-04

7. AUTHOR(s): Eric Beyer, Peter Buneman

8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0440

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Decision Sciences
The Wharton School
Philadelphia, PA 19104

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Task NR049-360

11. CONTROLLING OFFICE NAME AND ADDRESS:
Office of Naval Research
Department of the Navy
800 N. Quincy St., Arlington, VA 22217

12. REPORT DATE: January 23, 1979

13. NUMBER OF PAGES: 11

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 14p.

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Dynamic memory allocation, FIFO allocation, message queueing, first-fit, best-fit

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

It is often desirable for programs to communicate by queueing messages in shared areas of secondary storage. An algorithm is presented for the allocation of variable size blocks which are to be freed in the same order that they are allocated. Unlike a wrap-around technique, it does not require an initial allocation of a fixed amount of memory, but exploits the ability of many operating systems to increase the available storage as needed. By means of a worst-case analysis, its space efficiency is shown to be optimal.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A Space Efficient Dynamic Allocation Algoritnm for Queueing Messages

Eric Beyer
Peter Buneman

Abstract

It is often desirable for programs to communicate by queuing messages in shared areas of secondary storage. A simple algorithm is presented for the allocation of variable size blocks which are to be freed in the same order that they are allocated. Unlike a wrap-around technique, this method does not require an initial allocation of a fixed amount of memory, but exploits the ability of many operating systems to increase the available storage as needed. By means of a worst-case analysis, its space efficiency is shown to be optimal.

Keywords and Phrases: Dynamic memory allocation, FIFO allocation, message queuing, first-fit, best-fit.

CR Categories: 3.81, 4.32, 4.35, 5.25

We shall describe a dynamic storage allocation algorithm for variable size blocks which are allocated and freed in first-in-first-out (FIFO) order. Such a situation may occur when one process queues varying length messages to an asynchronous receiving process using shared areas of primary or secondary storage. The usual method for allocating space for queued messages is to use a simple wrap-around technique (Knuth [2]) in a fixed length array. However, this method requires an a priori limit to be placed on the maximum total length of messages that may be queued. In order to take advantage of the ability of many operating

---

1. General Electric Co., Valley Forge Space Center, P.O. Box 8555, Philadelphia.
2. Department of Computer and Information Science, University of Pennsylvania, Pa. 19174.

systems to increase dynamically the storage area as required, one must resort to a more sophisticated storage allocation algorithm. The "first-fit" and "best-fit" methods are well known candidates for this algorithm. In first-fit, available blocks of storage are examined in the order of their starting addresses, and an allocation is placed in the first available block, or "hole", into which it will fit. In best-fit, the allocation request is placed into the smallest available hole into which it fits. The "two-hole" method described below offers the advantage of simplicity and, for FIFO allocation, a worst case efficiency which is no worse than that of first-fit and better than best-fit.

The measure of efficiency used here is related to a measure defined by Robson [3] who studied allocation strategies for general (not FIFO) allocation in bounded memory when the block sizes are restricted. Given a sequence of requests and a storage allocation algorithm, let A(t) be the total amount of store in use at time t (this depends only on the sequence) and let R(t) be the current memory limit or total amount of storage needed by the allocation algorithm to accommodate these blocks. For a given allocation algorithm and sequence of requests, we define the efficiency to be

$$\frac{\max_{0<t<T} A(t)}{\max_{0<t<T} R(t)}$$

where T is the duration of the sequence. The efficiency of the algorithm is the lower bound on this measure for all

finite sequences. For example, tne following simple FIFO sequence shows that the efficiency of first-fit and best-fit is no better than 1/2. a_i, d_i, s_i denote respectively the allocation time, deallocation time and size for the ith request for a block of storage.

i	a_i	d_i	s_i
1	0	2	n
2	1	4	1
3	3	5	1+n

The efficiency, $(n + 2)/(2n + 2)$, of both these algorithms for this sequence approaches 1/2 as $n \to \infty$. Best fit can be shown to have an efficiency which is strictly less than 1/2 and it can also be shown that no FIFO allocation algorithm can have an efficiency better than 1/2 (Beyer [1]). This gives some limited confirmation to the report on simulations by Shore [4] which indicate that best-fit can be relatively inefficient.

As its name implies, for FIFO allocation, the two-hole method never permits more than two holes of unoccupied storage. Informally, the allocation algorithm may be described as follows: when the ith request for a block B is made

(a) If there is just one hole that starts at the origin and will accommodate B, then place B at the origin

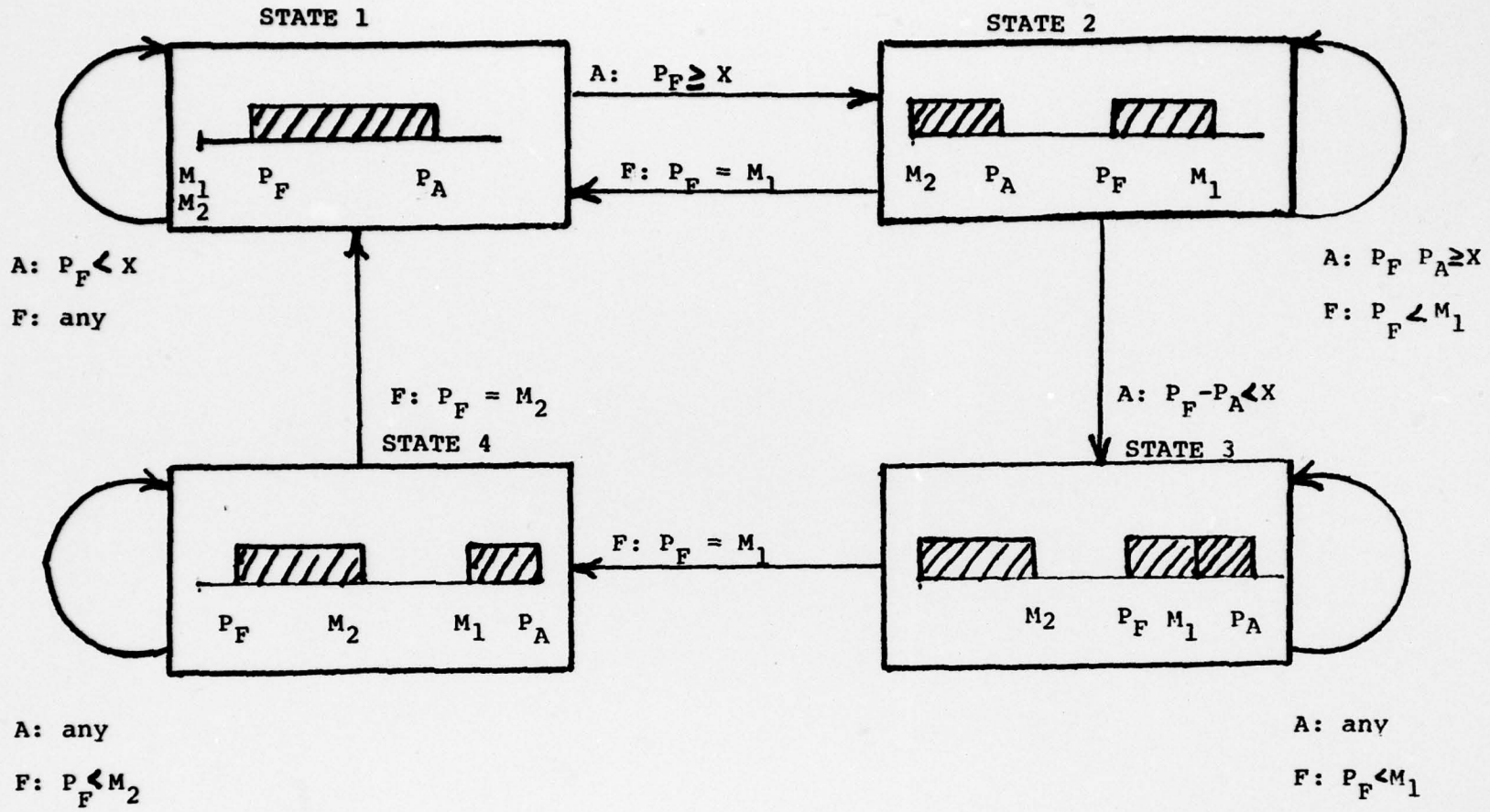

Figure 1. The four memory configurations of the two-hole algorithm. A: denotes the condition before a block of size X is allocated; F: the condition after a block has been freed.

(b) else if B may be placed immediately to the right of the previously allocated block then do that

(c) else place B to the right of the rightmost allocated block.

A formal description of the algorithm is given at the end of this paper. The diagram in figure 1 shows the four possible memory states that can occur. In this diagram P is the starting address of the next block to be freed and P is the right hand end of the most recently allocated block.

Initially, all pointers are set to 0 and memory is in state 1. New allocations are placed at P_A (rule b above) and blocks are freed from P_F until P_F exceeds X, the size of the block to be allocated. Then, by rule a, state 2 is entered. In state 2, M_1 marks the memory limit. New allocations are placed in the one hole (rule b). If a block will not fit into this hole, rule c causes state 3 to be entered. The other possibility is that P_F first catches up with M_1 and the system reverts to state 1. In state 3 allocations are always made at the current memory limit. M_2 marks the old position of allocation from state 2. In state 3 the blocks, in order, occupy the intervals $[P_F, M_1]$, $[0, M_2]$ and $[M_1, P_F]$. State 4 is entered when the allocated blocks in the interval $[P_F, M_1]$ have all been freed. There are two memory holes in this state and allocation continues at the memory limit. Subsequent freeing of blocks in

$[0, M_2]$ returns the system to state 1.

We now outline the proof that the efficiency of the two-hole method is no less than 1/2. We shall denote the current maximum allocated store by A* and the current maximum limit of memory required by R*; both A* and R* are monotone nondecreasing functions of time. Note that A*/R* can decrease only when the maximum memory limit R* increases through an allocation. The state transitions 1->2, 2->2, 2->1, 3->4 and 4->1 do not cause increases to the current memory limit and cannot decrease the efficiency. Of the remaining transitions, 2->3 is of central importance. Just after this transition, let X' be the size of the block allocated, A' be the resulting allocated memory and R' the resulting limit of memory. Also let $G' = P_F - M_2$ be the length of the one hole. Since allocation of X' forced the transition 2->3,

$$X' > G' \qquad (1)$$

Now $2A^* \geq 2A' = 2M_1 + 2X' - 2G'$

$$= R' + (M_1 - G') + (X' - G') > R ,$$

since the terms $(M_1 - G')$ and $(X' - G')$ are both positive. If this transition causes a new maximum in memory limit then $R = R^*$ so that the transition 2->3 cannot cause A*/R* to fall below 1/2.

After any allocation in the subsequent transitions 3->3 and 4->4, let A be the currently allocated memory.

Let L be the total additional

memory allocated since the 2->3 transition and let G be the total amount of memory freed. Suppose again that an allocation causes a new maximum, R*, in the memory limit. Then

$$A = M_1 + X' + L - G' - G \qquad (2)$$

and since the new allocations have all occurred at the right of $M_1 + X'$,

$$R^* = M_1 + X' + L \qquad (3)$$

and memory has been freed only between 0 and M_1,

$$G + G' < M_1 \qquad (4)$$

We consider two cases:

(i) $G \geq L$ (5)

which implies that $A' \geq A$, so that

$$\begin{aligned} 2A^* \geq 2A' &= 2M_1 + 2X' - 2G' \\ &> M_1 + X' + G + (X' - G') \quad \text{(by 4)} \\ &\geq R^* \quad \text{(by 3 and 5)} \end{aligned}$$

(ii) $G < L$ (6)

in which case

$$\begin{aligned} 2A^* &> 2A \\ &> M_1 + 2X' + 2L - (G' + G) \quad \text{(by 2 and 4)} \\ &> R^* \quad \text{(by 1, 3 and 6)} \end{aligned}$$

Thus allocations in the transitions 3->3 and 4->4 cannot bring A*/R* below 1/2. The only remaining transition is 1->1 which can be taken care of in a manner similar to 2->3; and this completes the proof.

It should be emphasized that the usefulness of this relatively simple method results from the storage management techniques available in many operating systems. As a point of practical interest, two-way communication between programs can be set up through two independent storage areas. Each program is privileged to write to one and read from the other. In a header, each program writes the values of the parameters P_A, P_F, M_1, M_2 for its write-privileged storage area and also indicates to the other program which message it last read. Thus each program frees and allocates in its write-privileged area and, provided reading and writing the header are indivisible operations, update anomalies cannot occur.

The Procedures for Allocation and Freeing

In the following it is assumed that the length of a message can be determined when it is freed. In practice, the length of each block is encoded at the beginning of the block or an end marker is used to determine the boundaries between blocks.

```
procedure ALLOC(X)
  if STATE = 1 and X < P  then
    begin
      STATE <- 2;
      M  <- P , P  <- X
    end
  else if STATE = 2 and  X > P  - P  then
    begin
      STATE <- 3;
      M  <- P ; P  <- M  + X;
    end
  else P  <- P  + X

procedure FREE;
  begin
    X <- size of block to be freed;
    P  <- P  + X;
    if P  = P  then
        begin
          P  <- 0; P  <- 0;
        end
    else if STATE = 2 and P  = M  then
      begin
        STATE <- 1;
        P  <- 0; M  <- 0;
      end
    else if STATE = 3 and P  = M  then
      begin
        STATE <- 4;
        P  <- 0;
      end
    else if STATE = 4 and P  = M  then
      begin
        STATE <- 1;
        P   <- M ; M  <- 0; M  <- 0;
      end
  end
```

References

1. Beyer, E. Moore School Report, University of Pennsylvania (1978).

2. Knuth, D.E. The Art of Computer Programming, Vol. 1 Fundamental Algorithms, 240-248, Addison Wesley, Reading, Mass., 1968

3. Robson, J.M. An Estimate of the Store Size Necessary for Dynamic Storage Allocation, J. ACM Vol. 18 3 (July 1971), 416-423.

4. Shore, J.E. On the External Fragmentation Produced by First-Fit and Best-Fit Allocation Strategies. Comm. ACM 18, 8 (August 1975), 433-440.

INSTRUCTIONS FOR PREPARATION OF REPORT DOCUMENTATION PAGE

RESPONSIBILITY. The controlling DoD office will be responsible for completion of the Report Documentation Page, DD Form 1473, in all technical reports prepared by or for DoD organizations.

CLASSIFICATION. Since this Report Documentation Page, DD Form 1473, is used in preparing announcements, bibliographies, and data banks, it should be unclassified if possible. If a classification is required, identify the classified items on the page by the appropriate symbol.

COMPLETION GUIDE

General. Make Blocks 1, 4, 5, 6, 7, 11, 13, 15, and 16 agree with the corresponding information on the report cover. Leave Blocks 2 and 3 blank.

Block 1. Report Number. Enter the unique alphanumeric report number shown on the cover.

Block 2. Government Accession No. Leave Blank. This space is for use by the Defense Documentation Center.

Block 3. Recipient's Catalog Number. Leave blank. This space is for the use of the report recipient to assist in future retrieval of the document.

Block 4. Title and Subtitle. Enter the title in all capital letters exactly as it appears on the publication. Titles should be unclassified whenever possible. Write out the English equivalent for Greek letters and mathematical symbols in the title *(see "Abstracting Scientific and Technical Reports of Defense-sponsored RDT/E," AD-667 000)*. If the report has a subtitle, this subtitle should follow the main title, be separated by a comma or semicolon if appropriate, and be initially capitalized. If a publication has a title in a foreign language, translate the title into English and follow the English translation with the title in the original language. Make every effort to simplify the title before publication.

Block 5. Type of Report and Period Covered. Indicate here whether report is interim, final, etc., and, if applicable, inclusive dates of period covered, such as the life of a contract covered in a final contractor report.

Block 6. Performing Organization Report Number. Only numbers other than the official report number shown in Block 1, such as series numbers for in-house reports or a contractor/grantee number assigned by him, will be placed in this space. If no such numbers are used, leave this space blank.

Block 7. Author*(s)*. Include corresponding information from the report cover. Give the name*(s)* of the author*(s)* in conventional order *(for example, John R. Doe or, if author prefers, J. Robert Doe)*. In addition, list the affiliation of an author if it differs from that of the performing organization.

Block 8. Contract or Grant Number*(s)*. For a contractor or grantee report, enter the complete contract or grant number*(s)* under which the work reported was accomplished. Leave blank in in-house reports.

Block 9. Performing Organization Name and Address. For in-house reports enter the name and address, including office symbol, of the performing activity. For contractor or grantee reports enter the name and address of the contractor or grantee who prepared the report and identify the appropriate corporate division, school, laboratory, etc., of the author. List city, state, and ZIP Code.

Block 10. Program Element, Project, Task Area, and Work Unit Numbers. Enter here the number code from the applicable Department of Defense form, such as the DD Form 1498, "Research and Technology Work Unit Summary" or the DD Form 1634, "Research and Development Planning Summary," which identifies the program element, project, task area, and work unit or equivalent under which the work was authorized.

Block 11. Controlling Office Name and Address. Enter the full, official name and address, including office symbol, of the controlling office. *(Equates to funding/sponsoring agency. For definition see DoD Directive 5200.20, "Distribution Statements on Technical Documents.")*

Block 12. Report Date. Enter here the day, month, and year or month and year as shown on the cover.

Block 13. Number of Pages. Enter the total number of pages.

Block 14. Monitoring Agency Name and Address *(if different from Controlling Office)*. For use when the controlling or funding office does not directly administer a project, contract, or grant, but delegates the administrative responsibility to another organization.

Blocks 15 & 15a. Security Classification of the Report: Declassification/Downgrading Schedule of the Report. Enter in 15 the highest classification of the report. If appropriate, enter in 15a the declassification/downgrading schedule of the report, using the abbreviations for declassification/downgrading schedules listed in paragraph 4-207 of DoD 5200.1-R.

Block 16. Distribution Statement of the Report. Insert here the applicable distribution statement of the report from DoD Directive 5200.20, "Distribution Statements on Technical Documents."

Block 17. Distribution Statement *(of the abstract entered in Block 20, if different from the distribution statement of the report)*. Insert here the applicable distribution statement of the abstract from DoD Directive 5200.20, "Distribution Statements on Technical Documents."

Block 18. Supplementary Notes. Enter information not included elsewhere but useful, such as: Prepared in cooperation with . . . Translation of *(or by)* . . . Presented at conference of . . . To be published in . . .

Block 19. Key Words. Select terms or short phrases that identify the principal subjects covered in the report, and are sufficiently specific and precise to be used as index entries for cataloging, conforming to standard terminology. The DoD "Thesaurus of Engineering and Scientific Terms" *(TEST)*, AD-672 000, can be helpful.

Block 20. Abstract. The abstract should be a brief *(not to exceed 200 words)* factual summary of the most significant information contained in the report. If possible, the abstract of a classified report should be unclassified and the abstract to an unclassified report should consist of publicly- releasable information. If the report contains a significant bibliography or literature survey, mention it here. For information on preparing abstracts see "Abstracting Scientific and Technical Reports of Defense-Sponsored RDT&E," AD-667 000.

☆U.S.Government Printing Office: 1974—603-617/3162 3-1